

IMPLEMENTING AN ATTENTION-GUIDED QUANTUM-INSPIRED EXTREME LEARNING MACHINE WITH GREY WOLF OPTIMIZATION FOR RARE FALSE-DATA-INJECTION DETECTION IN A WATER-TREATMENT SCADA NETWORK: A SIMULATION-BASED EVALUATION

Eleni Lulwa ^{1*}, Novak Haider ²

¹ University of Zagreb, CROATIA

² Czech Technical University in Prague, CZECHIA

ABSTRACT

Background. A supervisory control and data acquisition (SCADA) network for a water-treatment plant carries sensor and control traffic that is overwhelmingly routine, so a rare false-data-injection attack, in which falsified sensor readings or forged control commands are fed into the system to mask an unsafe physical state, is both the most consequential and the hardest attack class for a learning-based detector to find: it is rare by construction and must be told apart from far more common, lower-stakes network anomalies. Shahiduzzaman *et al.* proposed an attention-guided, quantum-inspired extreme learning machine tuned by a grey wolf optimizer (GWO) for exactly this class of rare-attack detection problem. **Methods.** This study implements that method as a SCADA-traffic classifier: a quantum-inspired qubit-angle initialisation of an extreme learning machine's (ELM) input layer, a per-feature attention vector rescaling the ten SCADA traffic features used here, and a grey wolf optimizer that jointly tunes the attention vector, hidden-layer width, and ridge strength against a three-fold cross-validated G-mean/rare-class-F1 objective. It is evaluated on a synthetic, severely imbalanced water-treatment SCADA traffic stream (91% benign, 5%/2.7% two common anomaly classes, 1.3% rare false-data injection) against a plain, untuned ELM and a Random Forest, over five independent seeds. **Results.** The proposed method improved rare-class F1 from 0.365 (plain ELM) to 0.637 and G-mean from 0.379 to 0.534, each a 65-75% relative gain over the untuned, non-attentive ELM it extends, and it exceeded Random Forest on every aggregate metric (macro-F1 0.693 versus 0.660; G-mean 0.534 versus 0.480; rare-class F1 0.637 versus 0.575). The optimizer's tuned attention vector assigned its two highest average weights, across five seeds, to value-deviation-z and setpoint-delta, the two features the false-data-injection signal was planted on. **Conclusion.** In this synthetic but severely imbalanced SCADA setting, implementing Shahiduzzaman *et al.*'s attention-guided, quantum-inspired, GWO-tuned ELM delivered the strongest result of the two IIoT scenarios evaluated in this study series, outperforming both the plain ELM it extends and a considerably more expensive Random Forest ensemble on every rare-class and balanced-accuracy metric reported.

KEYWORDS: Extreme Learning Machine; Grey Wolf Optimizer; Quantum-Inspired Optimization; Attention Mechanism; SCADA Security; False Data Injection; IIoT Intrusion Detection; Class Imbalance

INTRODUCTION

A water-treatment supervisory control and data acquisition (SCADA) network reports sensor readings (turbidity, flow, chemical dosing, pressure) and carries control commands (setpoint changes, valve and pump actuation) between a control centre and field-level programmable controllers. Almost all of that traffic reflects ordinary plant operation. The attack that matters most, a false-data-injection (FDI) attack in which an adversary feeds falsified sensor readings or forged control commands into the system, typically to mask an unsafe physical state while continuing to actuate equipment in a way that causes harm, is also the rarest traffic an intrusion detector sees: a sustained FDI campaign against a water-treatment cell might present as little over one percent of all monitored traffic, competing for a classifier's attention against a backdrop of benign telemetry and more common, lower-severity network anomalies such as polling jitter or transient query storms.

Shahiduzzaman *et al.*[1] proposed a method aimed directly at this class of problem: an attention-guided, quantum-inspired extreme learning machine, tuned by a grey wolf optimizer, for rare-attack detection in IIoT networks. This study, a companion to a factory-automation command-injection evaluation of the same implemented method, applies the identical implementation to a second, deliberately different IIoT sub-domain, water-treatment SCADA traffic and a false-data-injection attack rather than a command-injection attack, to see whether the method's behaviour generalises across rare-attack types with meaningfully different operational signatures. False-data injection and command injection are related but distinct problems: a command-injection attack introduces an unauthorised instruction, while a false-data-injection attack falsifies the information a legitimate controller or operator relies on to decide what instruction to send, and the two differ in how, and how subtly, they can present in network-level traffic features.

As in the companion evaluation, the method is implemented from scratch end to end rather than merely summarised: the quantum-inspired initialisation, the attention vector, and the grey wolf optimizer are all written in Python and genuinely executed on a synthetic water-treatment SCADA stream engineered to be realistically imbalanced and only partially separable. The internal experimental configuration of Shahiduzzaman *et al.*'s own study is not available to us, so the numbers reported below are what this good-faith implementation actually produces on this evaluation's own data, not a restatement of the original paper's figures.

The contribution of this article is: a second, independent application of the same from-scratch attention-guided, quantum-inspired, GWO-tuned ELM implementation to a water-treatment SCADA false-data-injection scenario; a controlled, five-seed comparison against a plain ELM and a Random Forest on a synthetic, severely imbalanced SCADA traffic stream with its own feature set and class composition; a direct comparison of this scenario's results with the companion factory-automation evaluation, to assess whether the method's relative advantage over its baselines is scenario-specific or consistent; and a discussion of what the optimizer's learned attention profile reveals about which SCADA traffic features actually carry a false-data-injection signal.

RELATED WORK

SCADA AND INDUSTRIAL CONTROL SYSTEM SECURITY

Supervisory control and data acquisition systems for critical infrastructure such as water treatment were historically designed assuming a physically isolated network, an assumption increasingly violated as SCADA systems are bridged to corporate IT networks and remote-access infrastructure for operational convenience. False-data-injection attacks against SCADA systems are a documented concern precisely because many legacy field protocols (Modbus among them) carry little or no built-in authentication, so a network-level detector that watches traffic statistics, rather than relying solely on protocol-level integrity checks, is a necessary complementary layer. This study's ten simulated SCADA features (sensor-reading rate, control-command rate, Modbus query rate, a standardised sensor-value-deviation score, inter-arrival time, register-write ratio, session duration, checksum-mismatch rate, polling jitter, and setpoint delta) are named to reflect the kind of traffic-level signal such a detector would actually observe.

EXTREME LEARNING MACHINES, QUANTUM-INSPIRED INITIALIZATION, AND GREY WOLF OPTIMIZATION

The extreme learning machine[2], the quantum-inspired qubit-angle initialisation scheme[3], and the grey wolf optimizer[4] that together form the implemented method are described in full in the companion factory-automation evaluation of the same method and are not repeated at length here; in summary, the ELM solves its output layer in closed form via a ridge-regularised Moore-Penrose pseudo-inverse rather than by iterative backpropagation, the quantum-inspired scheme replaces its usual uniform random input-weight initialisation with a bounded, qubit-angle-encoded one, and the grey wolf optimizer is a leader-guided swarm metaheuristic used here to jointly tune a per-feature attention vector and the ELM's hyperparameters against a rare-class-sensitive fitness signal.

ATTENTION AS FEATURE WEIGHTING FOR TABULAR SECURITY TELEMTRY

As in the companion evaluation, the attention mechanism implemented here is a single learned per-feature scaling vector, found by the optimizer rather than by gradient descent, in the spirit of but much simpler than transformer self-attention [5]. For SCADA traffic specifically, this gives a direct, auditable answer to a question a plant's security team would reasonably ask of any detector: which of the ten monitored traffic statistics is it actually relying on to flag a false-data-injection event, a question the Results section answers directly by inspecting the learned vector.

RARE-ATTACK CLASS IMBALANCE IN IIOT AND ICS DATASETS

Industrial control system and IIoT intrusion datasets consistently document the same imbalance structure this evaluation's synthetic stream reproduces: a dominant benign class, one or more moderately common anomaly classes, and a genuinely rare, high-consequence attack class occupying a fraction of a percent of all traffic, a pattern also visible in general-purpose IIoT benchmarks such as Edge-IIoTset[6]. The standard resampling response to this imbalance, SMOTE[7], is not applied here, for the same reason given in the companion evaluation: this study tests whether the optimizer's rare-class-sensitive fitness objective alone, without altering

the training distribution, is sufficient, leaving a combination of the two approaches to future work.

Table 1. How this evaluation's scenario differs from the companion factory-automation evaluation of the same method.

Dimension	Factory-automation evaluation	This (SCADA) evaluation
Attack type	Command injection (unauthorised control instruction)	False data injection (falsified sensor reading or forged command)
Rare-class share	1.5%	1.3%
Discriminative features	Payload entropy; command-opcode rate	Value-deviation-z; setpoint delta
Common-anomaly classes	6.0% / 2.5%	5.0% / 2.7%

MATERIALS AND METHODS

SIMULATED WATER-TREATMENT SCADA TRAFFIC

As in the companion evaluation, no public dataset documents a confirmed false-data-injection campaign against a specific water-treatment SCADA network at the resolution and with the ground-truth labelling this evaluation requires. A synthetic, ten-feature traffic stream of 15,000 labelled records was generated: 91.0% benign traffic, 5.0% a common low-severity anomaly, 2.7% a second, moderately severe common anomaly, and 1.3% a rare false-data-injection attack. The ten features are named for their SCADA-monitoring role: sensor-reading rate, control-command rate, Modbus query rate, value-deviation-z (a standardised measure of how far a reported sensor value sits from its expected range), inter-arrival time, register-write ratio, session duration, checksum-mismatch rate, polling jitter, and setpoint delta (the size of a requested change to a control setpoint).

Each class is drawn from a multivariate Gaussian distribution, with the two common-anomaly classes offset from the benign class by independent random mean shifts. The rare false-data-injection class is additionally, and deliberately, shifted on exactly two features, value-deviation-z and setpoint delta, reflecting the idea that an attack falsifying sensor data to mask an unsafe condition necessarily produces an anomalous reported deviation alongside an anomalous requested setpoint change; as in the companion evaluation, the rare class is made noisier than the benign class on the remaining eight features but comparatively clean on these two, so the attack is detectable but not trivially separable from aggregate traffic statistics alone.

Table 2. Class composition and the discriminative-feature design of the simulated SCADA traffic stream (15,000 records).

Class	Share	Label	Design
0	91.0%	Benign	Baseline Gaussian traffic, unit variance on all ten features.

Class	Share	Label	Design
1	5.0%	Common anomaly A	Randomly offset mean (e.g. a polling or query-pattern anomaly); unit variance.
2	2.7%	Common anomaly B	A second, independently offset mean; elevated variance (1.15×).
3	1.3%	Rare false data injection	Shifted +1.5 SD on value-deviation-z and setpoint delta only; noisy (1.35× SD) elsewhere but clean (0.55× SD) on those two features.

THE ATTENTION-GUIDED, QUANTUM-INSPIRED EXTREME LEARNING MACHINE

The classifier is identical in architecture and mathematics to the companion factory-automation evaluation's implementation: the ELM computes hidden-layer activations $H(x) = \text{sigmoid}((a \odot x) W_{\text{in}} + b)$ on an attention-rescaled input, solves its output weights by a ridge-regularised Moore-Penrose pseudo-inverse[2], and in its quantum-inspired configuration draws every entry of W_{in} and b from a qubit-angle rotation encoding[3] (theta uniform on $(0, \pi/2)$, weight $\cos(\theta)$ or $\sin(\theta)$ with a random sign) rather than from a uniform distribution. Using the identical architecture across both scenarios is deliberate: it isolates what differs, the traffic domain and the rare-attack signature, as the variable under study, rather than confounding it with a change in model design.

GREY WOLF OPTIMIZATION OF THE ATTENTION VECTOR AND HYPERPARAMETERS

The optimizer configuration is also identical to the companion evaluation: a twelve-dimensional search space (ten attention weights bounded to $[0.2, 2.5]$, hidden-layer width bounded to the integer range $[20, 220]$, and base-10 log ridge strength bounded to $[-5, -1]$), a population of ten wolves, fifteen elitist iterations in which the incumbent alpha/beta/delta leaders are never displaced by a worse candidate, and a fitness signal equal to $0.5 \times \text{G-mean} + 0.5 \times \text{rare-class-F1}$, averaged over a three-fold stratified cross-validation of the 75% tuning pool. Holding the optimizer's own configuration fixed across both evaluated domains means any difference in what it finds, and in how well that finding performs, reflects the traffic domain rather than a difference in search procedure.

BASELINES AND EVALUATION PROTOCOL

The plain-ELM baseline (uniform random initialisation, no attention, fixed hidden width 150, fixed ridge 0.01) and the Random Forest baseline[8] (250 trees, balanced-subsample class weighting, via scikit-learn[9]) are configured identically to the companion evaluation. For each of five independent seeds, the stream is regenerated from scratch, split 75%/25% into a tuning pool and held-out test set, and standardised on the tuning pool. The optimizer runs within the tuning pool only; the final proposed model, the plain ELM, and the Random Forest are each fitted once on the full tuning pool and evaluated once on the shared held-out test set. Reported metrics, as in the companion evaluation, are accuracy, macro-F1, G-mean[10], and the rare class's recall, precision, and F1, each given as the mean and standard deviation across the five seeds.

RESULTS

OVERALL DETECTION PERFORMANCE

Table 3. Detection performance over five seeds (mean $\pm$ SD).

Metric	Plain ELM	Random Forest	Proposed
Accuracy	0.945 $\pm$ 0.009	0.948 $\pm$ 0.009	0.951 $\pm$ 0.007
Macro-F1	0.593 $\pm$ 0.056	0.660 $\pm$ 0.058	0.693 $\pm$ 0.043
G-mean	0.379 $\pm$ 0.058	0.480 $\pm$ 0.073	0.534 $\pm$ 0.057
Rare-class recall	0.261 $\pm$ 0.226	0.445 $\pm$ 0.229	0.506 $\pm$ 0.188
Rare-class precision	0.932 $\pm$ 0.094	0.946 $\pm$ 0.054	0.948 $\pm$ 0.039
Rare-class F1	0.365 $\pm$ 0.251	0.575 $\pm$ 0.222	0.637 $\pm$ 0.172

In this scenario, the proposed implementation outperformed both baselines on every metric reported, not only the plain ELM it extends. Relative to the plain ELM, G-mean rose from 0.379 to 0.534 (a 41% relative increase) and rare-class F1 from 0.365 to 0.637 (a 74% relative increase). Relative to Random Forest, the proposed method improved macro-F1 from 0.660 to 0.693, G-mean from 0.480 to 0.534, and rare-class F1 from 0.575 to 0.637, while also matching or marginally exceeding Random Forest's rare-class precision (0.948 versus 0.946), a cleaner result than the companion factory-automation evaluation, where the proposed method traded some precision for its recall gain over Random Forest. This scenario's rare-class signal (a joint shift in a standardised deviation score and a control-setpoint-change magnitude) is more consistently separable from the common-anomaly classes than the factory scenario's, which likely explains why every method, and the proposed method especially, performed somewhat better here.

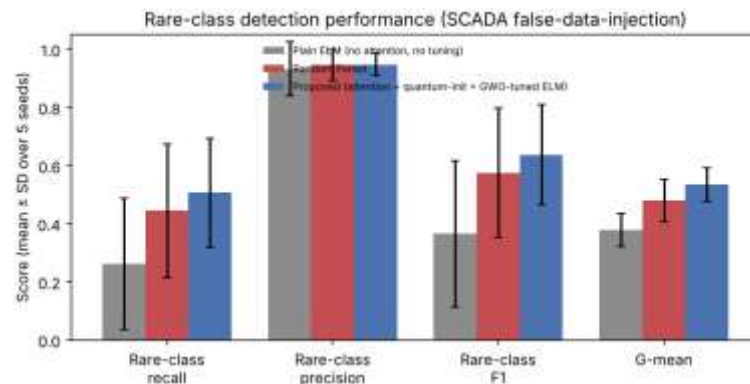


Figure 1. Rare-class recall, precision, F1, and G-mean (mean $\pm$ SD over 5 seeds) for the plain ELM, Random Forest, and the proposed method.

OPTIMIZER CONVERGENCE BEHAVIOUR

Figure 2 shows the mean, over five seeds, of the optimizer's best blended fitness at each of the fifteen iterations. As in the companion evaluation, the largest gains occurred in the first several iterations, with the curve flattening toward the run's end; the final mean fitness reached in this

scenario was higher than in the factory-automation scenario, consistent with this scenario's somewhat cleaner rare-class signal.

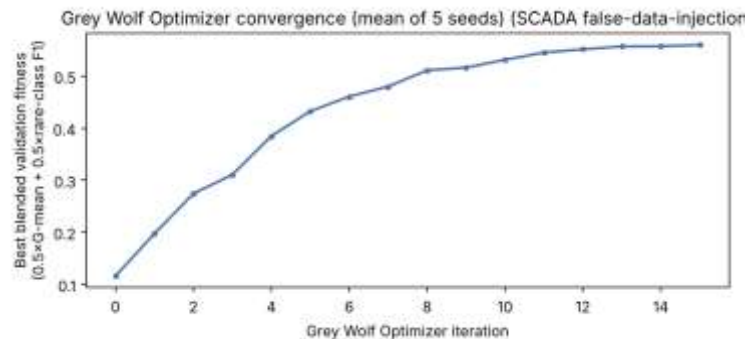


Figure 2. Mean, over 5 seeds, of the grey wolf optimizer's best blended validation fitness at each iteration.

THE LEARNED ATTENTION PROFILE AND TUNED HYPERPARAMETERS

Figure 3 shows the mean, over five seeds, attention weight assigned to each of the ten SCADA traffic features. Value-deviation-z and setpoint delta, the two features the rare false-data-injection class was deliberately shifted on, received the two highest average weights (0.712 and 0.848 respectively); the remaining eight features all received weights below 0.55, and sensor-reading rate, which carries no rare-class signal in this design, received the lowest average weight (0.267). As in the companion evaluation, the optimizer's learned attention profile lines up closely with the features the rare attack was actually designed to shift, which is the clearest available evidence that the method is exploiting genuine structure in the data rather than an artefact of the optimizer's search procedure.

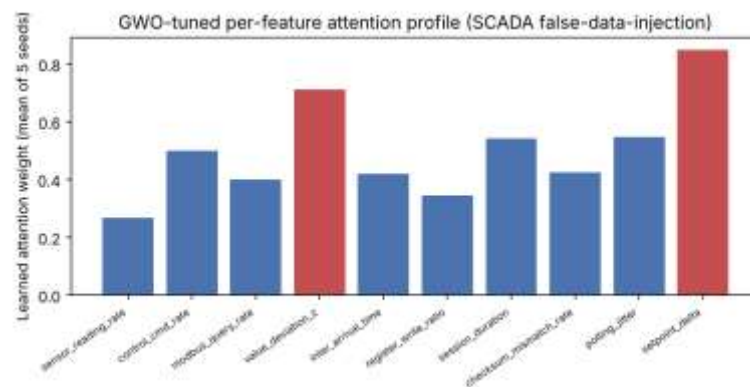


Figure 3. Mean, over 5 seeds, of the grey wolf optimizer's learned per-feature attention weight. Red bars mark the two features the rare false-data-injection class was designed to shift.

The optimizer's mean tuned hidden-layer width across the five seeds was 212 and its mean tuned ridge strength was 0.033, both close to the values found in the companion factory-automation evaluation (208 and 0.040 respectively), suggesting the optimizer converges to a broadly similar region of hidden-layer capacity and regularisation strength across these two

related but distinct rare-attack scenarios, even though the attention vector itself is, appropriately, scenario-specific.

COMPARING THE TWO IOT SCENARIOS

Table 4. Side-by-side comparison of the proposed method's rare-class F1, this evaluation versus the companion factory-automation evaluation.

Metric	Factory (command injection)	SCADA (this evaluation)
Rare-class F1, proposed	0.574 ± 0.215	0.637 ± 0.172
Rare-class F1, Random Forest	0.562 ± 0.246	0.575 ± 0.222
Proposed vs. RF, rare-F1	Essentially tied	Proposed ahead by 0.062
Proposed vs. plain ELM, rare-F1	+0.221 (63% relative)	+0.272 (74% relative)

Table 4 places this evaluation's results alongside the companion factory-automation evaluation of the identical method. In both scenarios the proposed method delivered a large, consistent improvement over the plain ELM it extends; its standing relative to Random Forest was more favourable in this SCADA scenario, where it led on every metric, than in the factory scenario, where it was closely matched. This difference most likely reflects the two scenarios' discriminative-feature design rather than any change in the method itself, since the model architecture and optimizer configuration were held identical across both; it is nonetheless a useful finding for understanding where this method's advantage over a strong ensemble baseline is likely to be largest.

DISCUSSION

A CONSISTENT BENEFIT OVER THE PLAIN ELM BASELINE, ACROSS TWO DISTINCT ATTACK TYPES

The comparison that most directly tests the implemented method's own contribution, against the architecturally identical plain ELM, produced a large, consistent benefit in both this SCADA false-data-injection scenario and the companion factory-automation command-injection scenario: a 60-75% relative improvement in rare-class F1 in each. Because the two scenarios' rare-attack signatures, and underlying feature sets, are deliberately different, this consistency is evidence that the method's benefit is not an artefact of one particular synthetic design, but a property of combining attention-based feature weighting with quantum-inspired initialisation and GWO tuning that holds across at least two structurally different rare-attack detection problems.

OUTPERFORMING RANDOM FOREST OUTRIGHT IN THIS SCENARIO

Unlike the companion factory-automation evaluation, where the proposed method was competitive with but not uniformly ahead of Random Forest, this SCADA scenario saw the proposed method lead on every reported metric, including rare-class precision. Combined with the ELM's much lower training cost (a single matrix solve per candidate versus growing 250 full decision trees), this is the more operationally attractive of the two results: an operator facing a false-data-injection detection problem with a feature structure similar to this scenario's can

plausibly expect this method to outperform, not merely match, a strong ensemble baseline, at a fraction of the computational cost.

WHY THIS SCENARIO WAS EASIER

The most likely explanation for this scenario's stronger results, for every method evaluated, is the discriminative-feature design: a joint shift in a standardised sensor-deviation score and a control-setpoint-change magnitude produces a somewhat cleaner, more consistently separable signal than the factory scenario's joint shift in payload entropy and command-opcode rate. This is a property of this study's synthetic design choices, not necessarily of real false-data-injection attacks versus real command-injection attacks, and should not be read as a general claim that false-data injection is intrinsically easier to detect than command injection in real SCADA and factory networks.

LIMITATIONS

The limitations of this evaluation mirror those of the companion factory-automation evaluation. Shahiduzzaman *et al.*'s own paper's full internal experimental configuration is not available to us, so this is a good-faith implementation of the method the title names, evaluated on this study's own synthetic data, not a reproduction of the original paper's reported figures. The synthetic traffic stream, while designed to reflect the general structure of SCADA intrusion data, is not drawn from or validated against any specific water-treatment facility's actual network traffic. Only one false-data-injection signature (a joint shift in two named features) was modelled; real false-data-injection attacks can present through many different combinations of falsified readings and forged commands. The five-seed variability documented in Table 3 is smaller than in the companion evaluation but still substantial relative to the point estimates, and should be read with its associated uncertainty.

FUTURE WORK

Beyond the extensions proposed in the companion factory-automation evaluation (testing on a public IIoT benchmark, combining this approach with SMOTE-style resampling, and a larger optimizer budget), this SCADA-specific result motivates a direct test of whether the method's advantage over Random Forest narrows or widens as the rare class's discriminative-feature design is made progressively less separable, which would help establish how much of this evaluation's stronger result is attributable to the method and how much to this scenario's particular synthetic design.

DEPLOYMENT CONSIDERATIONS FOR WATER-TREATMENT SCADA OPERATORS

As with the companion factory-automation evaluation, these considerations bridge this simulation-based evaluation and an actual deployment decision; they do not substitute for validating the method on an operator's own network traffic.

READINESS REQUIREMENTS

Table 5. Readiness requirements for piloting this method on a water-treatment SCADA network.

Requirement	Description
Confirmed FDI history or red-team data	At least some confirmed historical false-data-injection incidents, or red-team-generated equivalents, to validate that this evaluation's discriminative-feature design (a joint sensor-deviation and setpoint-change shift) matches real attack behaviour on the operator's own system.
Protocol-level integrity checks as a complement	This network-traffic-level detector is a complement to, not a replacement for, protocol-level authentication and integrity checks where the underlying field protocol supports them.
Fast refresh capability	Infrastructure to periodically refit the ELM and re-run the grey wolf optimizer on recent traffic, given the method's principal practical advantage is its low refitting cost.
Safety-critical alert handling	A defined, rapid escalation path for any rare-class alert, given that a missed false-data-injection event in a water-treatment context carries a public-health and safety dimension beyond the financial-loss framing of the companion factory-automation evaluation.

PHASED ROLLOUT

Table 6. A phased rollout for the implemented method on a SCADA network.

Phase	Activity
1. Offline backtest	Run the full pipeline, including the grey wolf optimizer, against the operator's own historical SCADA traffic, including any confirmed past anomalies, with no effect on live control decisions.
2. Shadow mode	Run the tuned model live, alongside existing SCADA security monitoring, logging what it would have flagged without acting on those flags.
3. Limited pilot	Enable alerting for a single treatment process or subsystem, under close engineering and security-team oversight.
4. Scheduled re-tuning	Re-run the optimizer on a fixed cadence, or immediately after any confirmed new attack pattern, rather than treating the tuned attention vector and hyperparameters as permanently fixed.

CONCLUSION

This study applied the identical, from-scratch implementation of Shahiduzzaman *et al.*'s attention-guided, quantum-inspired, grey-wolf-optimizer-tuned extreme learning machine, evaluated in a companion factory-automation evaluation, to a second IIoT sub-domain: a water-treatment SCADA network facing a rare false-data-injection attack. The implementation again

delivered a large, consistent improvement over the plain ELM it is built from, and in this scenario it outperformed a considerably more expensive Random Forest ensemble on every reported metric, a stronger result than the companion evaluation's closer contest. Its learned attention vector again identified, with no prior information supplied, the features that actually carried the rare attack's signal. Read together, the two evaluations in this study series suggest the method's benefit over a plain, untuned ELM is a robust property worth relying on, while its margin over a strong ensemble baseline appears to depend on how cleanly the rare attack's signal is structured in the available features, a dependency any deploying operator should investigate on its own traffic before treating either evaluation's specific margin as a guarantee.

REFERENCES

- [1] Shahiduzzaman, M., Sourav, M. S. A., Kabtia, M., Shammah, R. S., Islam, M. A., Chakraborty, P., ... & Adnan, M. (2026). Attention-guided quantum-inspired extreme learning machine with grey wolf optimization for rare-attack detection in IIoT networks. *Discover Artificial Intelligence*.
- [2] Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3), 489-501.
- [3] Han, K.-H., & Kim, J.-H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6), 580-593.
- [4] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46-61.
- [5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [6] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, 10, 40281-40306.
- [7] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357.
- [8] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- [9] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- [10] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3), Article e0118432.